3D Detector Response Calculations and Wire-Cell Prototype and Toolkit and LArSoft Integration

Brett Viren

Physics Department

BROOKHAVEN NATIONAL LABORATORY

SBU-BNL Touch Base
4 Oct 2016

Outline

Detector Field Response

LArTPC detector field response in an ideal[1] nutshell:

- Ar electrons are ionized and drift toward anode wire planes.
- Electron drift paths are nominally perpendicular to wire planes.
- Within a few pitch distances (∼cm) the pattern of wires significantly distorts the nominally uniform field.
- At the same distance scale the drifting electrons induce measurable currents on nearby wires.
- Current waveforms are readout with shaping and digitizing electronics.
- Noise happens.

The size and shape of the induced currents depend strongly on details on the scale of 0.1μ and 0.1mm.

[1]With some complicating and important reality ignored.

Two Scales of Granularity

Two scales matter:

simulation
: point response convolved over energy depositions (*eg.* Geant4 hits) of each simulated event. Need field response calculation on paths defined on a $\sim$0.1mm grid.

reconstruction
: response averaged over a uniform charge distribution filling each "wire region" ($\pm\frac{1}{2}$ pitch around a wire) and going out to $\pm \sim$3cm.

Response Depends on Two Electrostatic Fields

$\vec{E}_{weight,i}$ a constructed field for each *wire of interest* i

- wire i placed at 1V, all other electrodes at 0V.
- this is a consequence of "reciprocity"

$\vec{E}_{drift}$ a real, electrostatic field arising from the applied high voltage.

- Nominally chosen to obtain a desired drift velocity, which is driven in part by electron lifetime in LAr and max drift distance.
- Wire plane bias voltages are chosen to obtain desired "transparency" of each plane to the passing of drifting electrons.

The Response: Induced Currents

Induced current on a wire I_i in response to drifting charge q:

Shockley-Ramo:

$$I_i(t) = q\vec{v}_q(t) \cdot \vec{E}_{weight,i}$$

Drift velocity:

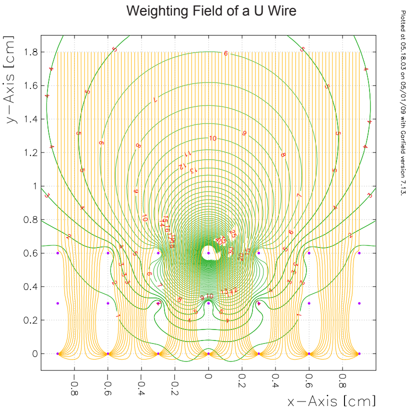
$$\vec{v}_q = \mu(|E_{drift}|, T) \times \vec{E}_{drift}$$

Given static $\vec{E}_{drift}$ and $\vec{E}_{weight,i}$:

- Induced current in a wire depends only on the **drift path**.
- Drift path depends only on its starting point.

$\rightarrow$ Charge diffusion makes the bookkeeping “challenging”.

2D Field Calculations



Equipotential weight (green) and drift paths (orange), Bo Yu using Garfield

- **F**inite **E**lement **M**ethod, high precision over limited 2D region.
- Reproduces major field features, especially away from wire planes.
- Relatively fast calculation, allows exploring (2D) parameter space.
- Used for LArSoft's simplistic simulation and signal reconstruction.
- Used Xiaoyue's improved simulation and Xin's improved signal reconstruction.

Why Calculate Fields in 3D?

- Some μBoone V-plane features seen in data possibly due to 3D wire structure.
- Generally validate 2D calculations and evaluate uncertainties.
- Explore inherently 3D, non-symmetric detector edge effects.
- Explore novel geometries more sensitive to 3D.
    - extra planes, hybrid collection/induction planes.

A New Field Calculation Method[2]

Switch from FEM → BEM: **Boundary Element Method**

- Difficult to scale FEM to 3D and required "large" volumes.
  → few mm scale: ∼days running
- BEM scales by electrode surface area.
  → few cm scale: ∼hours running

Almost user friendly software to do the calculations:

https://github.com/brettviren/larf

(LARF = LAr + Field)

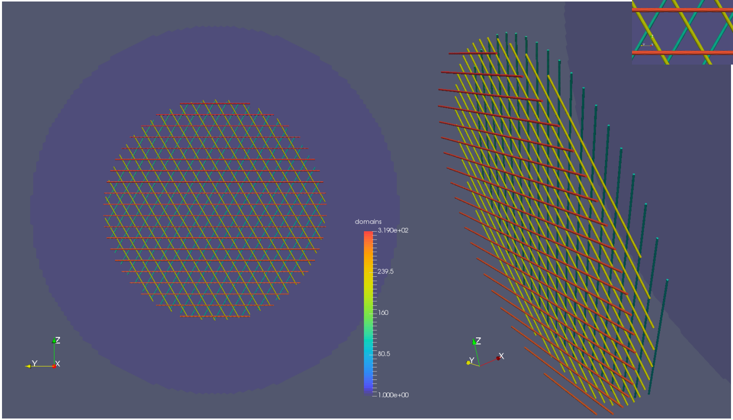
[2] To us.

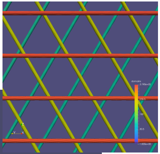
3D Field Calculation Procedure

1. Define **wire geometry**
2. Generate **surface mesh** on wires and other electrodes
3. Solve **surface boundary conditions** for each field
   - → again, one **drift** and one **weighting** per wire
4. Define electron drift path **starting points**
5. Step through **drift field** to produce **drift paths.**
6. Sample **weighting field** for a given wire, along a path to produce corresponding **current waveform.**
7. Repeat for many paths, **tabulate for simulation** and **form average for reconstruction.**

Some visualization of these steps ⟶

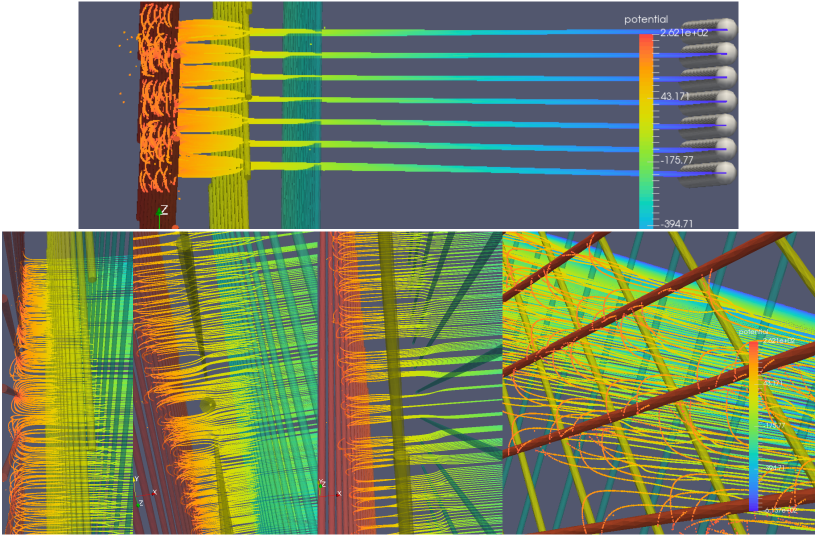
Surface Mesh



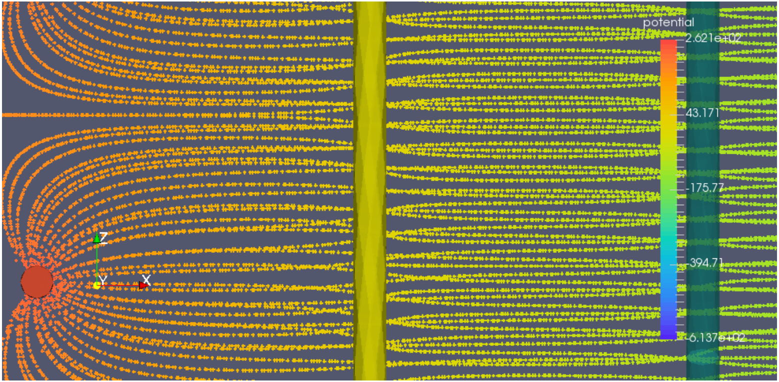


- Includes parameterized meshing for wires and simple shapes.
- Can roll-your-own or use *eg.* GMSH to generate your own.
- Mesh size drives accuracy and precision (and run time).

Drift Path Views

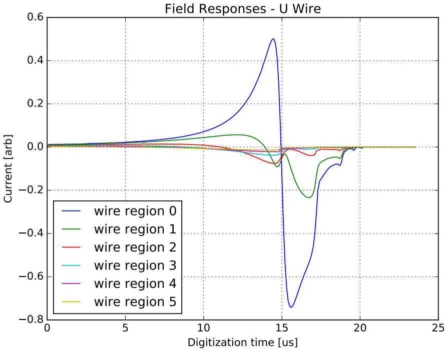


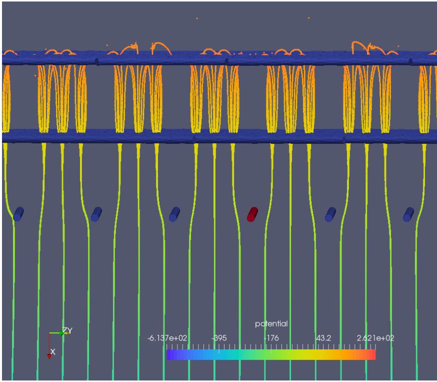
Stepping to Produce Paths



- Steps use 5th order Runge-Kutta with fixed step size (0.1μs).
- Each RK sub-step evaluates potential on 7 points to get gradient.
- Steps terminate if they "hit" a wire.

Coarse Response Functions - U-Wire

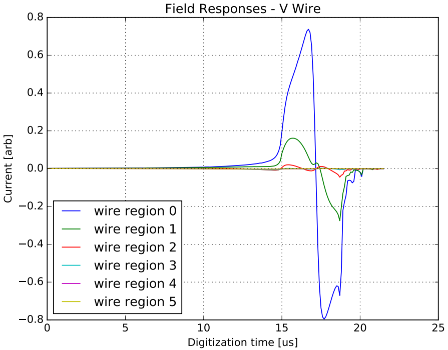


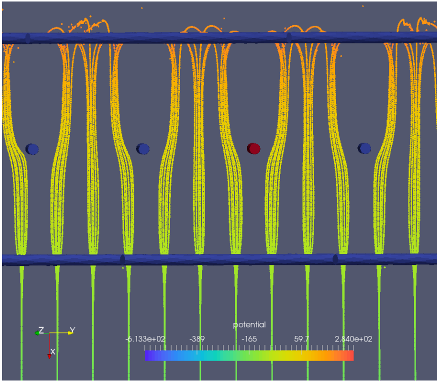


Average over 3mm in longitudinal direction and paths w/in $\pm$1.5mm of wire.

- Wire region 0 is $\pm$1.5mm around central U-wire.
- Wire regions 1-5 progressively further in transverse direction.
- Wire regions 6-9 not shown here.

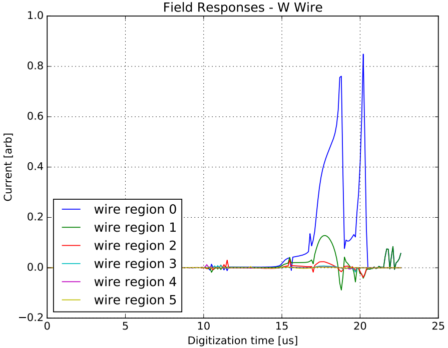
Coarse Response Functions - V-Wire

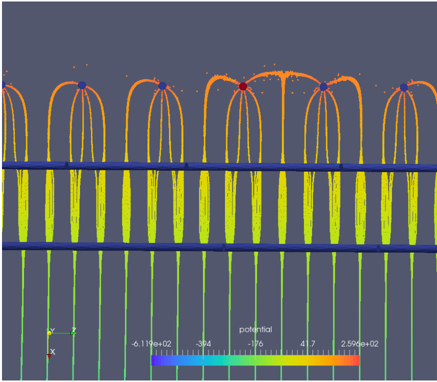




- Some end-of-track jaggies need checking, maybe due to "lucky symmetry".
  - → more severe example in W-wires

Coarse Response Functions - W-Wire

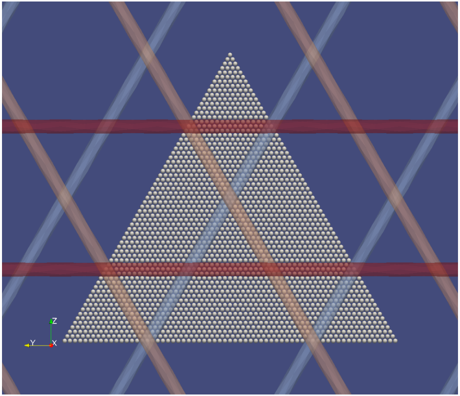




- One set of paths right on line of symmetry take an extra-long time.
  $\Rightarrow$ Need to fill in with more intermediate paths.
- Non-unipolar signals from charge in wire region ≥ 1

Choice of Paths for Fine Response Functions

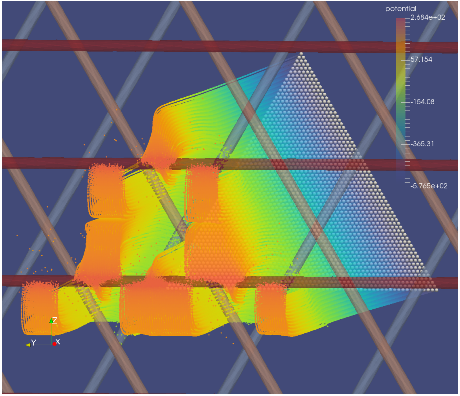
- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.



Starting points

Choice of Paths for Fine Response Functions

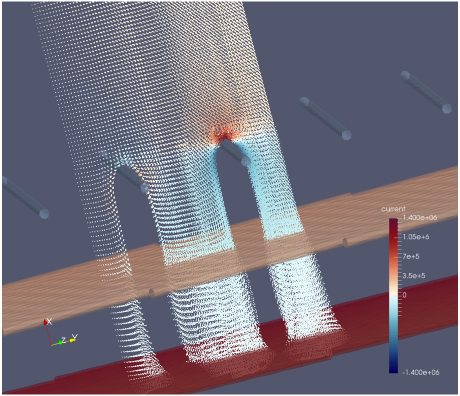
- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.



Paths colored by potential

Choice of Paths for Fine Response Functions

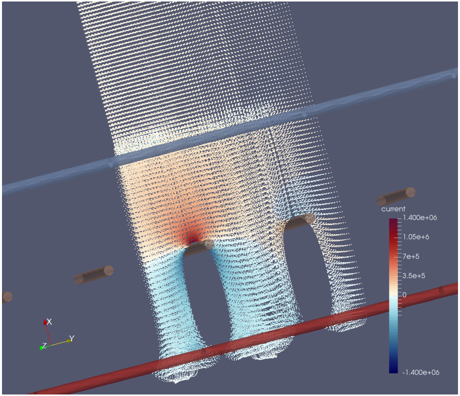
- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.



Paths colored by U-wire current

Choice of Paths for Fine Response Functions

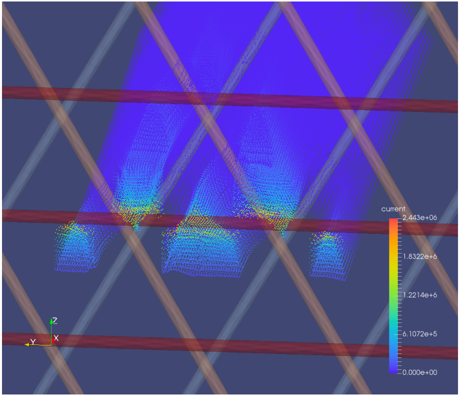
- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.



Paths colored by V-wire current

Choice of Paths for Fine Response Functions

- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.



Paths colored by W-wire current

Choice of Paths for Fine Response Functions

- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.

(t.b.d.)

Choice of Paths for Fine Response Functions

- Define region covering minimum unique patch of wire crossing pattern.
- Step through drift field for each start point to make path.
- For each drift path, sample $3N_{wires} = 30$ weighting fields. Really, reuse 3 boundary condition solutions, as can offset drift paths by $n \times pitch$.
- For simulation: map Gaussian-diffused energy deposition into triangle and convolve.
- For reconstruction: form averages over wire-regions.

(t.b.d.)

To Do List for Field Response

Roughly in order of priority

1. Finish the "t.b.d."s from the previous slide.
2. Implement DUNE wire patterns (trivial) and run LARF to catch up with the μBoone calculations (easy, but requires learning and some beefy workstation).
3. Evaluate uncertainties in simulation and signal reconstruction between use of 2D and 3D fields.
4. If major problems found, develop 3D simulation and signal reconstruction.
5. Look at detector edges, eg between two abutted APAs.
6. Look at novel wire geometry (4 plane, 2 collection planes, etc).

3D Detector Response Calculations

Wire-Cell Prototype and Toolkit and LArSoft Integration
- Prototype and Toolkit
- Integration

Wire-Cell Prototype and Toolkit

Prototype:

- Initial code structure, data model, build system.
- Emphasis is on the fast development of novel ideas.

Toolkit:

- More careful code structure and data model.
- Emphasis on long-term, multi-person development.
- Careful dependency management (concerned about single-user laptops, Linux clusters and up to HPC environments).
- Careful development of interfaces and layers for internal clarity, ease of integration, choices of code entry points.
- Adds a multi-threaded, data-flow programming paradigm option (still experimental).

Some novel dev still done in prototype, porting to tk ongoing, novel dev directly in tk is now an option.

Major Wire-Cell Features

- Real-world noise subtraction (μBoone).
- Waveform signal processing and simulation.
- 3D imaging of ionization activity (core Wire-Cell technique).
- 3D pattern recognition (tracks/showers).
- 3D final particle ID, energy reconstruction.

Except for the last, all exists at some level in prototype and some already exists in the toolkit.

LArSoft

- LArSoft = *art* **framework** + LArTPC simulation, reconstruction and other framework **modules and services**.
- Modules/services tend to either directly hold implementation code or call out to independent toolkits (eg, PANDORA and soon Wire-Cell).
- Used by most (all?) FNAL-based LArTPC experiments.
- Large, dedicated support team: 3-4(?) FNAL FTE, more if include *art* group and software build groups.

Wire-Cell / LArSoft Integration Strategy

- Wire-Cell is and will stay independent from LArSoft.
- Will follow David Adams lead of pushing *art* Services
  - Provide Wire Cell implementation of David's noise subtraction service.
  - Investigate David's simulation service and likely follow suit.
  - Naturally leverages the toolkit's Interface-oriented design.
- Integration code lives in `larwirecell` a package fully following the *Tao of Fermilab Software*.
- A UPS "product" `wirecell` holds built Wire Cell Toolkit binary libs.
  - Initial hack by me builds it on `fnal.gov` computers.
  - Getting cleaned up by FNAL experts (Lynn Garren).
- Initial target is **noise subtraction** (Brian Kirby).
  - Xin just finished "porting" this to the toolkit.
- Next is **signal processing** (??), then **simulation** (??).
  - Porting of signal processing to toolkit is just starting.